

IN THE CLAIMS:

1. (ORIGINAL) A parity storing method in an external storage subsystem comprising multiple disk drives including a plurality of storage blocks, comprising:
 - determining a number of the storage blocks to be used as data blocks and a number of the storage blocks to be used as parity blocks in a total number of storage blocks of the disk drives;
 - forming a three-dimensional block matrix of virtual data blocks corresponding to the determined number of the storage blocks to be used as the data blocks on Cartesian coordinates (X, Y, Z);
 - allocating virtual parity blocks to block planes related to the X, Y and Z-coordinates of the three-dimensional block matrix, respectively;
 - allocating the virtual data blocks and the virtual parity blocks to the storage blocks of the disk drives, respectively;
 - calculating parity information based upon data bits respectively stored in the storage blocks corresponding to the virtual data blocks of every block plane; and
 - storing the calculated parity information in the storage blocks corresponding to the virtual parity blocks for every block plane, respectively.
2. (ORIGINAL) The parity storing method according to claim 1, wherein the number of the storage blocks to be used as data blocks and the number of the storage blocks to be used as parity blocks are determined by calculating a maximum integer M satisfying $K \geq M^3 + 3M$, where K is the total number of storage blocks of the disk drives, so that the number of the data blocks is defined as M^3 and the number of the parity blocks is defined as 3M and the three-dimensional block matrix is a type of M x M x M.
3. (ORIGINAL) The parity storing method according to claim 1, wherein the allocating of the virtual data blocks and the virtual parity blocks to the storage blocks of the disk drives is performed by allocating the virtual parity block for each block plane to the storage block of the disk drive after the virtual data blocks belonging to each block plane are completely allocated to the storage blocks of the disk drives.
4. (ORIGINAL) The parity storing method according to claim 1, wherein the calculating of the parity information is performed by a bitwise operator performing exclusive OR (XOR) operation between the data bits stored in the storage blocks corresponding to the virtual data blocks of each block plane.

5. (ORIGINAL) An error block recovery method in an external storage subsystem comprising multiple disk drives including a plurality of storage blocks, comprising:
storing data and parity information by the parity storing method according to claim 1; and
recovering a plurality of error blocks by using parity blocks corresponding to the virtual parity blocks related to the respective block planes with the error blocks of X, Y and Z-coordinates.

6. (ORIGINAL) An error block recovering method in an external storage subsystem comprising multiple disk drives including a plurality of storage blocks, comprising:
storing data and parity information by the parity storing method according to claim 2; and
recovering a plurality of error blocks by using the parity blocks corresponding to the virtual parity blocks related to the respective block planes with the error blocks of X, Y and Z-coordinates.

7. (ORIGINAL) An error block recovering method in an external storage subsystem comprising multiple disk drives including a plurality of storage blocks, comprising:
storing data and parity information by the parity storing method according to claim 3; and
recovering a plurality of error blocks by using the parity blocks corresponding to the virtual parity blocks related to the respective block planes with the error blocks of X, Y and Z-coordinates.

8. (ORIGINAL) An error block recovering method in an external storage subsystem comprising multiple disk drives including a plurality of storage blocks, comprising:
storing data and parity information by the parity storing method according to claim 4; and
recovering a plurality of error blocks by using the parity blocks corresponding to the virtual parity blocks related to the respective block planes with the error blocks of X, Y and Z-coordinates.

9. (ORIGINAL) The error block recovering method according to claim 5, wherein the recovering the plurality of error blocks comprises:
counting the total numbers of the error blocks occurring in the block planes, respectively;
skipping the block planes with no error blocks or with two or more error blocks
performing an exclusive OR (XOR) operation between the data bits stored in the storage blocks, except for an error block, corresponding to the virtual data blocks belonging to each block plane with one error block;

recovering the one error block by comparing XOR operation results with the parity information stored in the storage block corresponding to the parity block for the block plane; and repeating the above recovering operations in regular order of the X, Y and Z-coordinates.

10. (ORIGINAL) The error block recovering method according to claim 6, wherein the recovering the plurality of error blocks comprises:

counting the total numbers of the error blocks occurring in the block planes, respectively;
skipping the block planes with no error blocks or with two or more error blocks each block plane with one error block;

recovering the one error block by comparing XOR operation results with the parity information stored in the storage block corresponding to the parity block for the block plane; and repeating the above recovering operations in regular order of the X, Y and Z-coordinates.

11. (ORIGINAL) The error block recovering method according to claim 7, wherein the recovering the plurality of error blocks comprises:

counting the total numbers of the error blocks occurring in the block planes, respectively;
skipping the block planes with no error blocks or with two or more error blocks
performing an exclusive OR (XOR) operation between the data bits stored in the storage blocks, except for an error block, corresponding to the virtual data blocks belonging to the block plane with one error block;

recovering the one error block by comparing XOR operation results with the parity information stored in the storage block corresponding to the parity block for the block plane; and repeating the above recovering operations in regular order of the X, Y and Z-coordinates.

12. (ORIGINAL) The error block recovering method according to claim 8, wherein the recovering the plurality of error blocks comprises:

counting the total numbers of the error blocks occurring in the block planes, respectively;
skipping the block planes with no error blocks or with two or more error blocks
performing an exclusive OR (XOR) operation between the data bits stored in the storage blocks, except for an error block, corresponding to the virtual data blocks belonging to the block plane with one error block;

recovering the one error block by comparing XOR operation results with the parity information stored in the storage block corresponding to the parity block for the block plane; and repeating the above recovering operations in regular order of the X, Y and Z-coordinates.

13. (ORIGINAL) A machine readable storage comprising multiple disk drives each

comprising a plurality of storage blocks, the storage comprising:

a programmed computer processor associating storage blocks of the disk drives to a virtual three-dimensional block matrix of virtual data blocks with virtual parity blocks allocated to each virtual data block plane, and controlling data input/output to/from the disk drives.

14. (ORIGINAL) The storage of claim 13, wherein the programmed computer processor further calculates parity information based upon the data bits respectively stored in the storage blocks corresponding to the virtual data blocks of each virtual data block plane.

15. (ORIGINAL) The storage of claim 13, wherein the programmed computer processor further recovers any number of error blocks in the storage blocks according to the virtual parity blocks corresponding to each virtual data block plane.

16. (ORIGINAL) The storage of claim 13, wherein the programmed computer processor generates the virtual three-dimensional block matrix by calculating a maximum integer M satisfying $K \geq M^3 + 3M$, where K is the total number of storage blocks of the disk drives, M^3 is a number of the storage blocks used as data blocks, and $3M$ is a number of the storage blocks used as parity blocks, and the three-dimensional block matrix is $M \times M \times M$.

17. (ORIGINAL) The storage of claim 15, wherein the programmed computer processor recovers the error blocks by repetitively:

counting a total number of the error blocks occurring in the virtual data block planes, respectively,

skipping the virtual data block planes with no error blocks or with two or more error blocks,

performing an exclusive OR (XOR) operation between the data bits stored in the storage blocks, except for an error block, corresponding to the virtual data blocks belonging to the virtual data block plane with one error block, and

recovering the one error block by comparing the XOR operation results with the parity information stored in the storage block corresponding to the virtual parity block for the virtual data block plane.

18. (ORIGINAL) The storage of claim 13, wherein the programmed computer processor associates the storage blocks of the disk drives to the virtual three-dimensional block matrix by:

determining a number of the storage blocks to be used as data blocks and a number of

the storage blocks to be used as parity blocks in a total number of storage blocks of the disk drives,

generating the three-dimensional block matrix of the virtual data blocks corresponding to the determined number of the storage blocks to be used as the data blocks on Cartesian coordinates (X, Y, Z),

allocating the virtual parity blocks to the virtual data block planes related to the X, Y and Z-coordinates of the three-dimensional block matrix, respectively,

allocating the virtual data blocks and the virtual parity blocks to the storage blocks of the disk drives, respectively,

calculating parity information based upon the data bits respectively stored in the storage blocks corresponding to the virtual data blocks of every virtual data block plane, and

storing the calculated parity information in the storage blocks corresponding to the virtual parity blocks for every virtual data block plane, respectively.

19. (ORIGINAL) A machine readable storage system, comprising:
a redundant array of independent disks (RAID) comprising disk drives each having a plurality of storage blocks;
a host computer processing data and requesting data input/output of the disk drives;
a controller in communication with the host computer and the RAID, and controlling the RAID operations according to a process of associating storage blocks of the disk drives to a virtual three-dimensional block matrix of virtual data blocks with virtual parity blocks allocated to each virtual data block plane.

20. (ORIGINAL) The system of claim 19, wherein the controller further calculates parity information based upon the data bits respectively stored in the storage blocks corresponding to the virtual data blocks of each virtual block plane.

21. (ORIGINAL) The system of claim 19, wherein the controller further recovers any number of error blocks in the storage blocks according to the virtual parity blocks corresponding to each virtual data block plane.

22. (ORIGINAL) A method, comprising:
allocating storage blocks of disk drives according to a virtual three-dimensional block matrix of virtual data blocks and virtual parity block planes corresponding to each virtual data block plane to store data and parity information in the storage blocks of the disk drives; and
using the virtual parity plane and the corresponding virtual data block plane to recover

any number of error blocks in the corresponding allocated storage blocks.

23. (ORIGINAL) A machine readable storage comprising multiple disk drives each comprising a plurality of storage blocks, the storage comprising:

a programmed computer processor associating storage blocks of the disk drives to a virtual three-dimensional block matrix of virtual data blocks with virtual error information blocks allocated to each virtual data block plane, and recovering any number of error blocks in the storage blocks according to the error information blocks corresponding to each virtual data block plane.

24. (WITHDRAWN)